**Object Oriented Development Lab Sheet 2**

**INTRODUCTION TO LINQ**

**Note: For this lab we are using Console Apps – select .NET Framework as template type**

**Graphical user interface

Description automatically generated with low confidence**

**Exercise 1**: Use LINQ to retrieve information from an array of numbers

This exercise will use the query syntax version of LINQ to retrieve a subset of numbers from an array of integers

Graphical user interface, text, application

Description automatically generated

**Exercise 2:** We want the same result here but using Lambda syntax. You will repeat the code from above but change the query to that shown below

**Graphical user interface, text, application

Description automatically generated**

**Exercise 3** : Use LINQ to retrieve file information. You will need to create a new class file called MyFileInfo which is shown below. You will also need to include using System.IO in your program. The program retrieves file information from your computer. It then uses LINQ to query this information, makes use of the new class you have created to result in an IEnumberable<MyFileInfo> which you can iterate through and display information.

Graphical user interface

Description automatically generated

**Graphical user interface, text, application, email

Description automatically generated**

**Table

Description automatically generated with medium confidence**

**Exercise 4:** An anonymous type. This program is almost the same as the program in exercise 3 above. The difference here is that you are not using a class such as MyFileInfo but you are utilising an anonymous type. You are defining the properties of this type/class in the select statement where you assign a value to Name, Length and CreationTime.

Graphical user interface, text, application

Description automatically generated

**Exercise 5**: Change the query from Exercise 4 to use Lambda syntax

**Graphical user interface, text

Description automatically generated**

**Exercise 6:** Deferred Execution

This question demonstrates deferred execution. You use a method called DoubleIt which doubles a number and also outputs a message. The method is called from the query but it does not output the message when called. This is because it is deferred until such time as it is used. This is demonstrated as the “Before the foreach loop” message displays before the method is executed.

Graphical user interface, text

Description automatically generated

**Exercise 7:** Chained queries. This can be useful for building queries. Start with something simple and add to it. On the left is query syntax. On the right it the lambda syntax. Use both in your code for practice but comment out one.

Text

Description automatically generatedGraphical user interface, application

Description automatically generated with medium confidence

**For the exercises below use both query syntax and also method syntax**

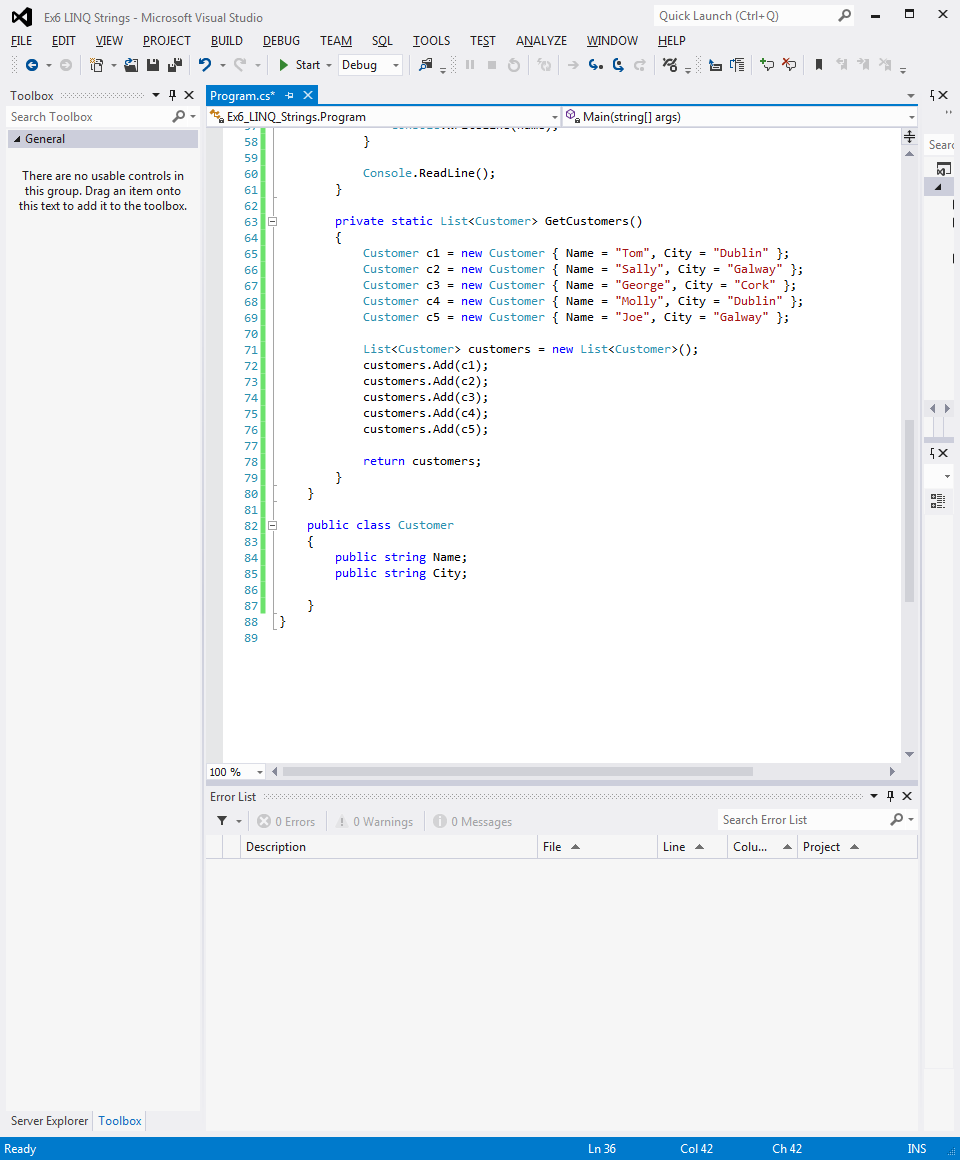
**Exercise 8:** Create an array of strings and create a simple LINQ query which displays the names to screen where the length of the name is greater than 4.

string[] names = {"Mary","Joseph","Michael","Sarah","Margaret","John" };

**Exercise 9:** Order the above alphabetically

**Exercise 10:** Include only names that begin with M

**Exercise 11:** Use the following code to create a list of customers. Create a LINQ query which extracts customer names where the City is Dublin



**Exercise 12:** Amend above to include Dublin or Galway names and order by name

**Exercise 13:** Return to your project idea and work on the XAML interface.